Jack Girard
ECE 637
Lab 2: 2D Random Processes

1. Power Spectral Density of an Image

1. Gray scale image *img04g.tif*

Figure 1: img04g.tif

2. Power spectral density plots for block sizes of 64 × 64, 128 × 128, and 256 × 256

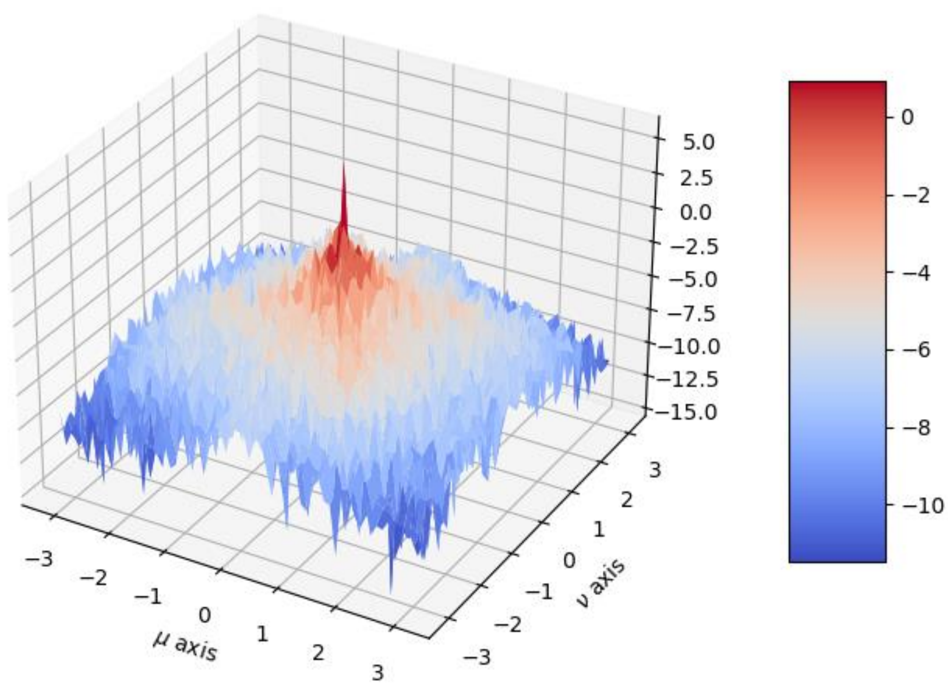

Figure 2: Power Spectral Density Plot – 64 × 64 Window Size

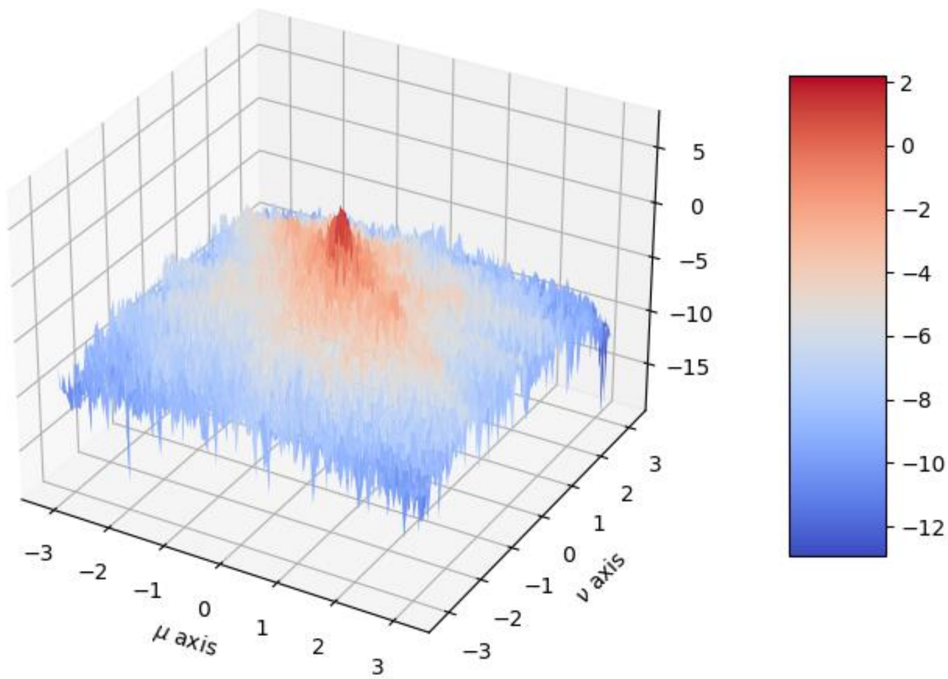

Figure 3: Power Spectral Density Plot – 128 × 128 Window Size

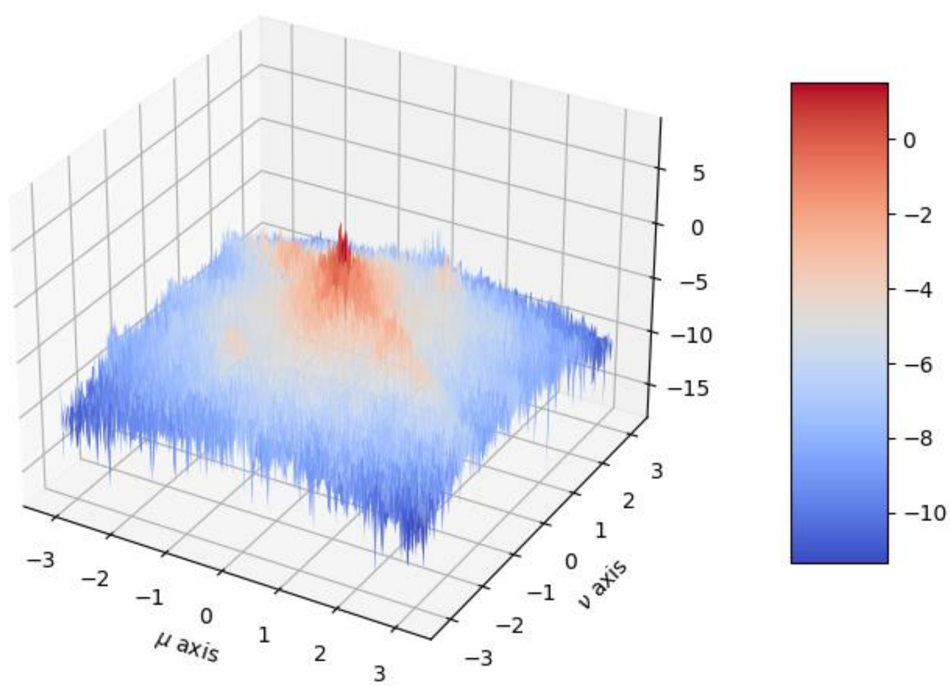

Figure 4: Power Spectral Density Plot – 256 × 256 Window Size

3. Improved power spectral density estimate

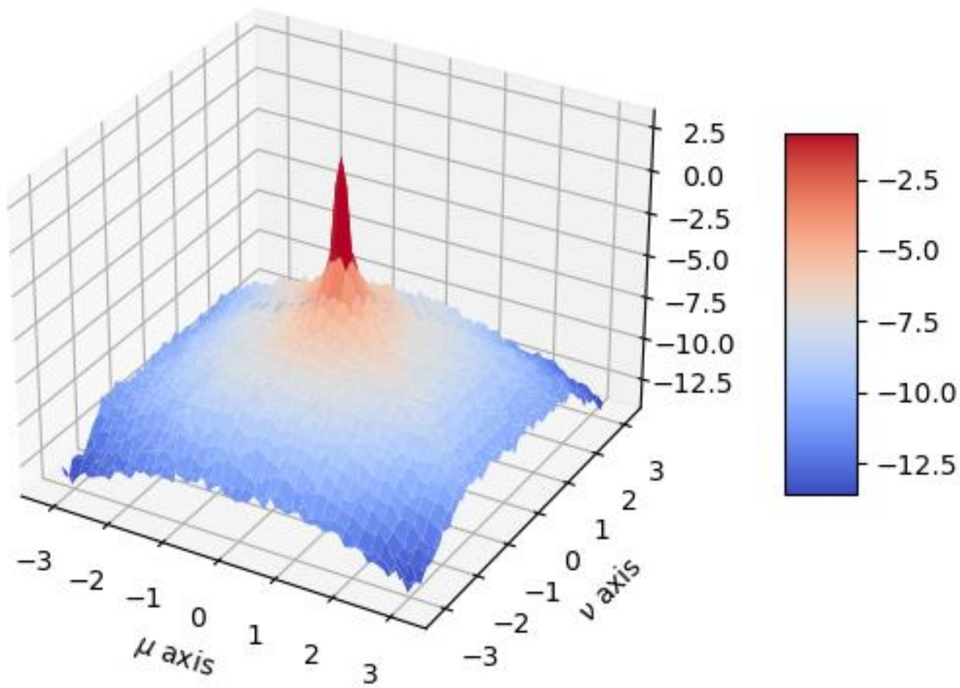

Figure 5: Improved Power Spectral Density Estimate

4. Code for *BetterSpecAnal(x)* function: *See Next Page*

```python
def BetterSpecAnal(x):
    # define window size
    N = 64
    # create hamming window
    W = np.outer(np.hamming(N), np.hamming(N))
    # calculate center of image in x and y directions
    cenx = len(x[0])/2
    ceny = len(x)/2
    # initialize empty matrix to store summation
    shape = (N,N)
    val = 0
    dt = x.dtype
    Zsum = np.empty(shape,dtype=dt)
    Zsum.fill(val)
    # define index set for i and j variables
    I = [-2, -1, 0, 1, 2]
    J = [-2, -1, 0, 1, 2]
    # nested for loop for each i and j index:
    for i in I:
        for j in J:
            # create new matrix y based on new window location
            y = x[int(cenx + (i-1)*N):int(cenx + i*N),\
                  int(ceny + (j-1)*N):int(ceny + j*N)]
            # multiply new window by Hamming window, store as z
            z = y*W
            # compute squared DFT magnitude
            Z = (1/N**2)*np.abs(np.fft.fft2(z))**2
            # use fftshift to move zero frequencies to center of plot
            Z = np.fft.fftshift(Z)
            # take the log of Z matrix
            Z = np.log(Z)
            # add current Z matrix to summation matrix
            Zsum += Z
    # once looping is completed, divide summation matrix by 25 to get the average
    Zavg = (1/25)*Zsum
    return Zavg
```

Figure 6: Jupyter Screenshot of BetterSpecAnal(x) Function Code

2. Power Spectral Density of a 2-D AR Process

1. Image 255 * (x + 0.5)

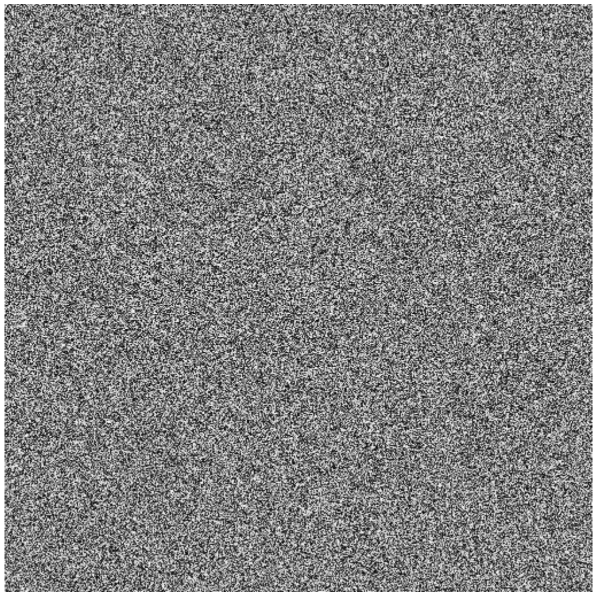

*Figure 7: Scaled x Image – 255 * (x + 0.5)*

2. Image y + 127

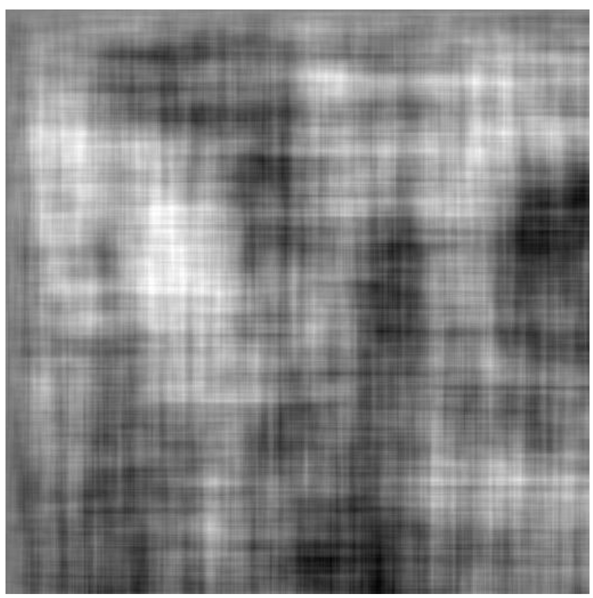

Figure 8: Filtered Image – y + 127

3. Mesh plot of the function $\log S_y\left(e^{j\mu}, e^{j\nu}\right)$

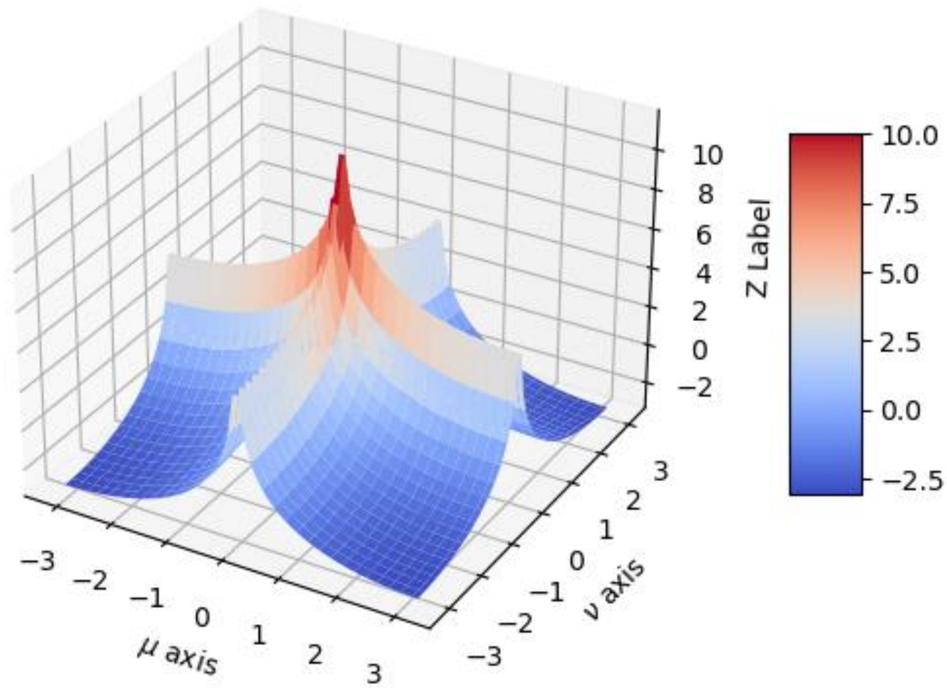

Figure 9: Mesh Plot of log $S_y\left(e^{j\mu}, e^{j\nu}\right)$

4. Mesh plot of the log of the estimated power spectral density using *BetterSpecAnal(y)*

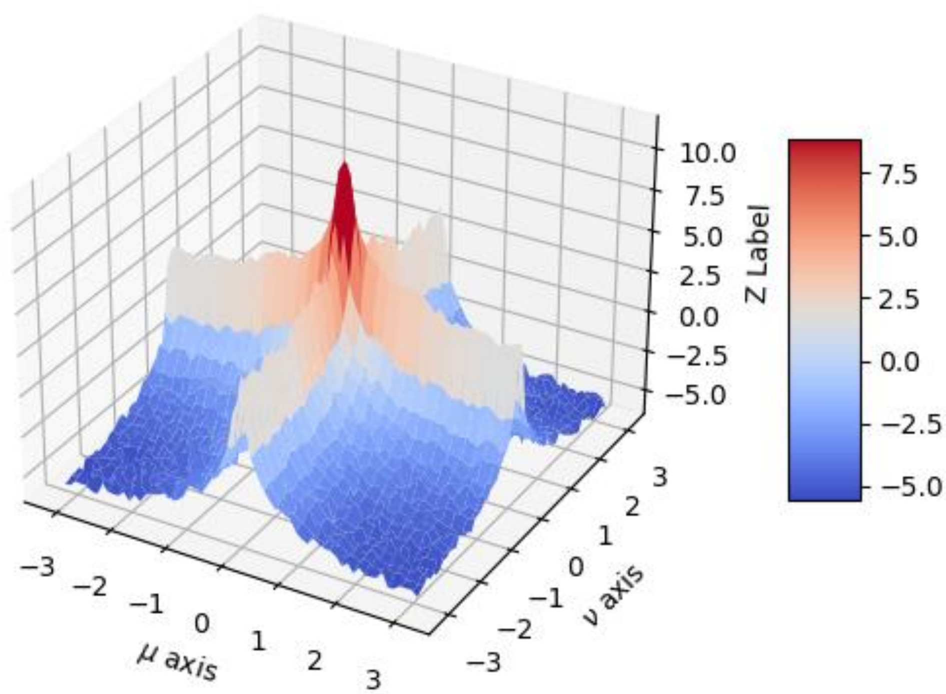

Figure 10: Mesh Plot of log of Estimated Power Spectral Density